FOREST SERVICE SIMULATOR (FSSIM) ©

TECHNICAL REPORT

Beta Release - February 1997

FOREST SERVICE SIMULATOR (FSSIM) ©

TECHNICAL REPORT

Beta Release

February 1997

The FSSIM software was developed by the B.C. Ministry of Forests, Timber Supply Branch. Documentation was prepared by Timberline Forest Inventory Consultants Ltd., with funding provided by Forest Renewal BC.

Copyright © 1997. B.C. Ministry of Forests

106-01618/

TABLE OF CONTENTS

INTRODUCTION 1
1.0 FSSIM DATA MODEL 3
2.0 FSSIM KEY MODELLING ASSUMPTIONS 9
3.0 PROCESS MODEL 17
APPENDIX I INTERNAL DATA RECORD STRUCTURE 31

LIST OF FIGURES

FIGURE 1.1 - FSSIM FOREST COMPONENTS 5
FIGURE 3.1 - MAIN PROGRAM FLOW CHART 18

INTRODUCTION

This report provides a technical description of the Forest Service Simulator (FSSIM). In distributing the FSSIM software, the Ministry of Forests (MoF) would like to encourage development of additional functionality to improve the future utility of FSSIM. In order to accomplish this, it is necessary to bridge the gap between the user and the FSSIM programmers. The objective in preparing this manual was to provide this bridge by instilling in the user an understanding of the assumptions inherent in the FSSIM data structure and processes.

The report is arranged in the following sections:

1. Data Model
2. Key Modelling Assumptions
3. Process Model

This Technical Report is designed to complement material contained in the FSSIM Users Manual. This latter document was designed to provide the user with a basic understanding of FSSIM background and concepts, and to assist in building data inputs and performing simulations. The user should consult the manual for information concerning:

1. Program installation
2. FSSIM conceptual description
3. Program functionality and flow
4. Running the program
5. Sample data set and simple analyses scenarios
6. "How to" Help pages
7. Glossary of terms

Note: While, for clarity, certain concepts illustrated in the Users Manual are also described in this document, it is assumed that the reader of this report is fully familiar with the contents of the Users Manual.

1.0 FSSIM DATA MODEL

1.1 ANALYSIS UNITS 4
1.2 ZONES 4
1.3 CLASSES 4
1.4 GROUPS 5
1.5 TRANSFERS 6
1.6 REGENERATION PARAMETERS 6
1.7 YIELD TABLES 6
1.8 YIELD TABLE ADJUSTMENT FACTORS (OAFS) 7
1.9 LANDBASE ADJUSTMENT FACTORS (LANDOAFS) 7
1.10 FUTURE ROAD ALLOWANCES 7
1.11 HARVEST TARGETS 7
1.12 HARVEST PRIORITIES 8
1.13 FOREST COVER CONSTRAINTS (FCCS) 8

1.0 FSSIM DATA MODEL

The underlying structure of FSSIM is built around the relationships between 3 components of the forest, namely: analysis units, zones and classes.

1.1 Analysis Units

An analysis unit (AU) represents an aggregation of forest classes with similar yield characteristics.

1.2 Zones

A zone is an aggregation of forest classes which share common management rules, such as forest cover constraints (FCCs).

1.3 Classes

A forest class represents components of the forest with similar attributes. FSSIM reads the starting forest inventory information from *class.dat*. Each record in the file contains specifications used in FSSIM to create classes. These specifications include zone, analysis unit, and age range. Age range is defined by lower and upper ages. FSSIM creates a class for every year within the age range specified. For example, an age range of 91-100 will result in the creation of 10 classes, with ages 91, 92,...100. Each class will contain $1/10^{th}$ of the area specified in the class file record.

Each class is also designated a class type which describes the way in which the class will contribute to the analysis. Class type must be O,I,D,N or blank.

Class Type	Harvest Contribution?	FCC Contribution?	Specify Reversion Age?
(O)perable	Yes	Yes	No
(I)noperable	No	Yes	No
(D)eferred	No	Yes	Yes
(N)on-contributing	No	No	Yes

Class types D and N revert to Operable when the reversion age is reached. Reversion age is only specified for class types D and N.

1.4 Groups

The Group is used in a more flexible fashion to aggregate these components for different purposes. FSSIM reads group data from the *group.dat* file, if one is present. Groups define areas bound together for analysis purposes. FSSIM allows two group types: (f)loat and (s)tatic. Area in a static group at the start of the planning horizon remains in that group throughout the entire planning horizon. On the other hand, a float group contains area that meets the group criteria at the beginning of the period. Float group constituency may change as a result of aging, harvest, regeneration, transfers etc., and is re-evaluated at the start of each period.

The relationships between all four components are shown in Figure 1.1.

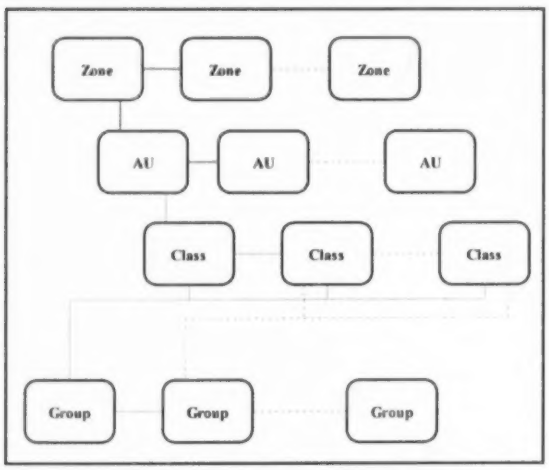

Figure 1.1 - FSSIM forest components

The initial relationships between these four components are established by the user in the FSSIM data model. Subsequent changes during the simulation are controlled by two data components; transfers and regeneration parameters.

1.5 Transfers

FSSIM reads zone or analysis unit transfers from the optional file *transfer.dat*. Transfer records identify changes to forest class attributes such as zone, AU, age, and Type. Transfers can be specified to occur during particular iterations.

1.6 Regeneration parameters

In the minimal FSSIM data model, classes are assigned to the same analysis unit following harvest, with no regeneration delay. Optionally, FSSIM reads regeneration assignments from *regen.dat*. The pre and post-regeneration analysis units are defined in *regen.dat*, as is the percentage of the harvested area moving to any post-regeneration analysis unit, and the regeneration delay to be used. If the *regen.dat* file is present, FSSIM cross checks regeneration assignment against the yield table list in the *vols.dat* file. All zones, analysis units and regenerating analysis units specified in the *regen.dat* file must also be specified in the *vols.dat* data file.

A number of additional data components are also specified by the user in the FSSIM data model. These are described in Sections 1.7 - 1.13.

1.7 Yield tables

Age-dependent yield lookup tables are specified in the *vols.dat* file. Yield tables may be derived from any external source. The only requirement is that they be defined as ordered pairs of volume/age values. Ages must fall between 0 and 350, and one yield table must be specified for each analysis unit. Ages do not have to be at regular intervals (example 10 years). In performing the lookup function, FSSIM uses a linear interpolation between intervals to determine volumes at specific ages. For a particular table, the last volume specified is applied to all classes older than that age. A volume of 0.0 for age 0 is implicit.

1.8 Yield table adjustment factors (OAFs)

Operational adjustment factors (OAFs) are used to scale yield table values. These are specified in the *oaf.dat* file and are used to adjust yields defined in *vols.dat*. In addition, the minimum harvest ages (MHA) for classes using each volume table are specified in this file. All tables specified in *vols.dat* should have a corresponding entry in *oaf.dat*. If not, the table uses default values of MHA = $-\infty$ and OAF = 1.00. The volume table number is analogous to AU number.

1.9 Landbase adjustment factors (LandOAFs)

FSSIM reads and applies land base adjustment factors from the *landoaf.dat* file. Land base adjustment factors are applied to zones and AUs, with analysis units taking priority. If zone is not specified then the OAF is applied to all the AU occurrences across all zones. If zone but not AU is specified then the OAF is applied to all AUs within the zone.

1.10 Future road allowances

Maximum age of classes assumed to be roaded and reductions for areas removed from the harvesting land base for future road development are specified in the *roads.dat* file. Forest classes equal to or less than the specified age at time zero will not have road reductions made to them. Other classes, after harvest by FSSIM, will be reduced in area by the road reduction factor. This reduction represents a permanent removal of area from the landbase. Forest cover constraint calculations exclude area removed for future roads..

1.11 Harvest targets

Annual harvest targets by period are read by FSSIM from the *targets.dat* file. Harvest target levels can be specified for ZONE or GROUP land units (Unit Type) and as VOLUME(m^3) or AREA(ha) levels (Target Type). Different target types can be employed in the same simulation. ZONE = 0 always refers to the entire land base.

Targets represent maximum volumes or areas to be harvested. Achieved levels may be less if particular forest cover constraints are limiting.

1.12 Harvest priorities

Zone, analysis unit and/or group harvest scheduling priorities by period are specified in the optional *priority.dat* file. Lower priority numbers have higher harvesting priority. The harvest priority mechanism is implemented by ordering the FSSIM harvest candidate list according to the specified priorities. All candidates assigned the highest priority will be harvested before any candidates of a lower priority, regardless of other factors such as age.

1.13 Forest cover constraints (FCCs)

Forest cover constraint (FCC) data is read from the file *axs.dat*. Constraints are imposed at either the zone or group level. Constraints are either a maximum area under a certain age (green-up), or a minimum area equal to or above a certain age (old-growth). Each definition consists of the time frame (in simulation periods) in which the constraints will be applied, the age thresholds and the limits expressed as percentages.

2.0 FSSIM KEY MODELLING ASSUMPTIONS

2.1 VOLUME@AGE LOOKUP 10
2.2 HARVEST@AGE / DISTRIBUTION OF HARVESTED AREA 10
2.3 CLASS FRAGMENTATION 11
2.4 LOOK-AHEAD 12
2.5 GROUP MEMBERSHIP 12
2.6 ROADING 13
2.7 ACCESS CONSTRAINT CALCULATIONS 13

2.0 FSSIM KEY MODELLING ASSUMPTIONS

2.1 Volume@age lookup

Volumes are derived using the class age at the beginning of the period.

Example:

class age = 93

period length = 10 years

In reality, harvest of the class could occur at any point during the 10 year period. Therefore the class age at harvest would be between 93-100. However, FSSIM derives the volume at age 93 for harvest calculations. Basing harvest volume on age at the beginning of the period is pessimistic, in that it does not account for growth during the period. In the previous version of FSSIM which employed class age ranges (ex. 93-100), the volume was derived using the upper limit of the class age range (ex. 100). MoF Timber Supply Branch staff are reviewing the implications of the current algorithm.

2.2 Harvest@age / distribution of harvested area

Currently, FSSIM assumes an even annual harvest across the period. That is, area harvested during a period is distributed evenly throughout the period.

Example:

harvest area = 5 000 ha

period length = 10 years

In reality, a specific area could be harvested at any point during the period. However, FSSIM assumes that 500 ha were harvested every year during the period. Other models assume that all area is harvested at the mid-point of the period. This current FSSIM algorithm is being reviewed as well.

To model the even annual area harvest assumption, FSSIM distributes area harvested during the period over the period length.

Area (ha)	Post-harvest Age	End of Period Age
500	-9	1
500	-7	3
500	-6	4
500	-5	5
500	-4	6
500	-3	7
500	-2	8
500	-1	9
500	0	10

If the algorithm were changed to harvest at the mid-point of the period, harvested area could not be distributed across the period length. In addition, post-harvest ages would be mid-pointed (ex. 5,10,15), and age-based forest cover constraints would lose precision.

2.3 Class Fragmentation

FSSIM timber classes are the smallest unit of area represented in the model, but they are not atomic. Class area may be broken up as a result of harvest, including partial harvest of the class area, as well as specification of multiple regenerating AUs. The FSSIM aggregation functionality, if activated, can result in the grouping of class fragments into larger classes, thereby reducing the size of the data structures and improving processing efficiency.

2.4 Look-ahead

Since FSSIM is a periodic model, it must account for changes in the state of the forest that occur during the period. Look-aheads are used to determine the state of the forest at the end of the period, for issues such as:

1. **Minimum harvest age**: Is current age + look-ahead ≥minimum harvest age?
2. **Group membership**: Will an class age into a group during the period? If so, it becomes a constituent of the group at the start of the period.
3. **Access constraints**: Is current age + look-ahead ≥ access constraint age?
4. **Transfer criteria**
5. **Reversions**

Look-aheads project to the end of the period assuming no harvest takes place.

2.5 Group Membership

Static group membership is assessed at time 0, and never changes. Area in a static group remains in that group throughout the planning horizon, regardless of any changes to the class which may occur.

Float group membership is reassessed at the start of each period in the planning horizon and reflects changes of state resulting from activities such as harvesting, regeneration, transfers, aging and reversion.

Because FSSIM is a periodic model, the state of the forest at the beginning of the period may be different from that at the end of the period. As a result, it is possible for classes to age into, or out of, a given group during the period.

Example:

Group	Age range
1	0-100
2	101-400

A class with age 93 at the beginning of the period would switch groups part way through the period. FSSIM assumes that the class belongs to both groups for the entire period, after which it would cease to be a member of Group 1.

2.6 Roading

FSSIM recognizes two types of roads, namely; existing and future. Existing roads are assumed to be in place at the beginning of the planning horizon, and are implemented in the model by specifying a maximum roaded age for every zone. Any classes assigned to that zone with an initialized age ≤ the specified roaded age are considered to have been roaded prior to the start of the planning horizon. Classes not meeting this criterion are eligible for roading at time of 1st harvest during the planning horizon. A class will only be roaded once during the simulation, regardless of how many times it is harvested.

The area removed by FSSIM during roading is permanently removed from the landbase. Therefore, the area of the zone or group(s) affected is permanently reduced. This is considered to be somewhat of a compromise. It was considered too binding to simply set roaded area to a permanently non-greenup status. On the other hand, it was considered too lax to allow roaded area to green up at the same rate as regenerating areas.

Because roading reduces the land area, it cannot occur during harvest, as it would affect access constraint calculations. FSSIM therefore performs roading after harvest and post-harvest constraint calculation steps.

2.7 Access Constraint Calculations

FSSIM allows two kinds of access constraints to be applied to a given zone or group.

1. Green-up: specified as a maximum allowable area < a given age, and
2. old-growth: specified as a minimum required area ≥ a given age.

Any given zone or group may have a number of green-up and/or old-growth constraints applied to it.

2.7.1 Green-up

Every zone and group maintains a counter indicating the current amount of non-green area. Maximum allowed harvest considering the green-up constraint is given by:

(zone or group area * green-up %) - current non-green area

This value is calculated for every active green-up constraint. As classes are harvested, the non-green area is incremented to reflect the addition of harvested area. Conversely, as classes age and reach the green-up age, the non-green are is decremented to reflect this.

A special case arises when green-up age < period length. In this case, a proportion of the area harvested during the period will green-up by period end, reducing the amount of non-green area and allowing further harvest. The proportion of harvested area that will green-up during the period is given by:

(period length - green-up age)/period length

Example:

period length = 10 years

green-up age = 7 years

(10-7)/10 = 30% of the harvested area will green-up by period end. This is accounted for when incrementing the non-green counter.

2.7.2 Old-growth

For the old-growth calculation, zones and groups maintain an age/area distribution. Two cases result:

Case 1. Excess old-growth

This occurs when the area ≥ the specified old-growth age using look-ahead exceeds the minimum required. In this case, harvesting can occur in the oldest classes until the requirement is met, or until only the minimum old-growth requirement remains.

Case 2. Insufficient old-growth

This occurs when there is insufficient area to meet the old-growth requirement. In this case, sufficient area in the oldest classes below the specified old-growth age is reserved from harvesting. Harvesting can then commence in classes younger than those reserved. This ensures that the old-growth requirement will be met as quickly as possible, without unnecessarily curtailing harvest activities.

3.0 PROCESS MODEL

3.1 MAIN PROGRAM FLOW PSEUDO CODE 18
3.2 TRANSFER 21
3.3 POPULATE (BEFORE) 22
3.4 DEVELOP HARVEST CANDIDATE LIST 24
3.5 REPORT (BEFORE) 25
3.6 HARVEST 25
3.7 REPORT CONSTRAINTS (AFTER) 27
3.8 ROAD 27
3.9 AGGREGATE 27
3.10 AGE 28
3.11 POPULATE (AFTER) 29
3.12 REPORT (AFTER) 29

3.0 PROCESS MODEL

This section details the main FSSIM process model and expands on each major component. Figure 3.1 provides a summary of program flow and major components. Each component is then discussed in the sections which follow. For each program component, documentation is in the form of a general description of the purpose and functionality of the component, followed by a pseudo code listing.

3.1 Main Program Flow Pseudo Code

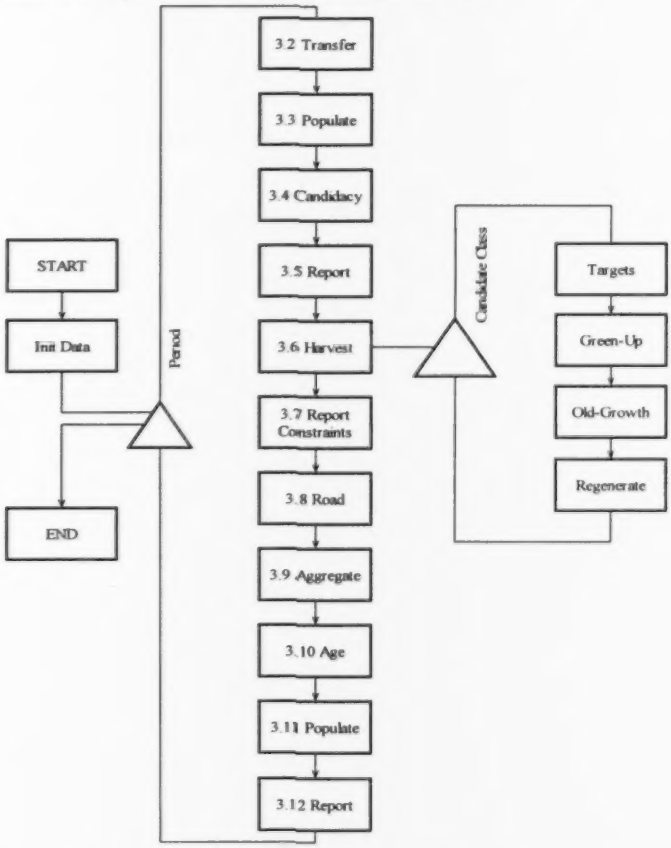

Figure 3.1 - Main program flow chart

START PROGRAM

Initialize Data:
FOR CurrentPeriod = 1 TO MaxPeriod
BEGIN CurrentPeriod

Transfer (see Section 3.2):
FOR EACH Class DO
BEGIN Class
Change timber class characteristics such as parent zone, parent AU, age and class type based on transfer parameters
END Class;

Populate(before) (see Section 3.3):
FOR EACH CLASS DO
BEGIN Class
Update all data structures to reflect conditions at the beginning of the period.
END Class;

Candidacy (see Section 3.4):
FOR EACH Class DO
BEGIN Class
Schedule classes for harvest based on harvest priorities, age and harvest rule; Create list of Candidate Classes.
END Class;

Report(before) (see Section 3.5):
Generate information for pre-harvest reporting

Harvest (see Section 3.6):

FOR EACH CandidateClass DO
BEGIN CandidateClass

Targets:

Calculate maximal area to harvest based on specified harvest targets. When more than one target applies to a given class, the target that returns the least area is the one that applies. This ensures that no targets are exceeded.

Green-up:

Calculate maximal area to harvest based on green-up requirements.

Old-Growth:

Calculate maximal area to harvest based on old-growth requirements.

IF HarvestArea > 0 DO
BEGIN DoHarvest

Regenerate;

Regenerate harvested area.

END DoHarvest;
END CandidateClass;

Report Constraints(after) (see Section 3.7):

Constraints reporting after harvest

Road (see Section 3.8):

FOR EACH HarvestedClass DO
BEGIN HarvestedClass

Update landbase to reflect roading; done here so as not to change landunit area during harvest which would affect access constraint calculations.

END HarvestedClass;

Aggregate (see Section 3.9):
FOR EACH Class DO
BEGIN Class
Aggregate like timber classes together for improved efficiency.
END Class;

Age (see Section 3.10):
FOR EACH Class DO
BEGIN Class
Age class to period end.
END Class;

Populate(after) (see Section 3.11):
FOR EACH CLASS DO
BEGIN Class
Update group and other data structures to reflect changes due to harvest, aging, etc. This is done for reporting purposes.
END Class;

Report(after) (see Section 3.12):
Post-harvest reporting.
END CurrentPeriod;

END PROGRAM

3.2 Transfer

The purpose of the transfer mechanism is to reassign forest class attributes, including zone, analysis unit, age and class type. If the *transfer.dat* file is included in the FSSIM data model, a transfer list will be created. At the start of each iteration, this list is checked to determine if any transfers are scheduled for that period.

```
BEGIN Transfer
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR EACH ZONE associated AU  DO
                BEGIN
                        FOR EACH AU associated CLASS  DO
                        BEGIN
                                If the zone/AU is in the Transferlist apply
                                transfer(change characteristics:  parent zone, parent
                                AU, age and classtype) to class
                        END;
                END;
        END ;
END;
```

3.3 Populate (before)

Prior to completing harvest and growth processes for the period, the FSSIM internal data structure is updated. This includes any changes brought about through transfers, or changes in floating group constituents.

```
BEGIN Populate Groups
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR ZONE associated AU  DO
                BEGIN
                        FOR AU associated CLASS  DO
                        BEGIN
                                Update Static and Float Group structures for class
                        END;
                END;
        END;
END ;
```

BEGIN **Populate Area**
FOR EACH Zone IN CurrentPeriod DO
BEGIN
FOR ZONE associated AU DO
BEGIN
FOR AU associated CLASS DO
BEGIN
Update static and floating group area by CLASS area
Update parent Zone by CLASS area
Update parent AU by CLASS area
END;
END;
END;
END ;

BEGIN **Populate Initial Age/Area Structures**
The Age/Area structure maintains an age/area distribution for the zone or group in question. This distribution is used for FCC calculations.

FOR EACH Zone IN CurrentPeriod DO
BEGIN
FOR Zone associated access constraint record DO
BEGIN
Create/Update Zone AGEAREA record based on access constraint requirements
END;
END;
FOR EACH Group IN CurrentPeriod DO
BEGIN
FOR Group associated access constraint record DO
BEGIN
Create/Update Group AGEAREA record
END;
END;
END ;

```
BEGIN Populate Age/Areas
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR ZONE associated AU  DO
                BEGIN
                        FOR AU associated CLASS  DO
                        BEGIN
                                Update Zone AgeArea record
                                Update Floating Group AgeArea record
                                Update Static Group AgeArea record
                        END;
                END;
        END;
END ;
```

3.4 Develop Harvest Candidate List

The FSSIM harvest rule is used to order forest classes for harvesting. In addition, if a *priority.dat* file is included in the FSSIM data model, classes are also ordered by specified combinations of zone, analysis unit and group. Priorities take precedence over harvest rules. For example, all classes with priority number n with be harvested before those with priority n+1.Classes with the same priority number are ordered according to the harvest rule. Therefore, the manner by which classes are scheduled depends upon the harvest rule chosen, and the priorities specified.

```
BEGIN Create Candidate List
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR ZONE associated AU  DO
                BEGIN
                        FOR AU associated CLASS  DO
                        BEGIN
                                Schedule timber classes for harvest based on harvest
                                priorities, and harvest rule.
                        END;
                END;
```

END;
END;

3.5 Report (Before)

Forest inventory characteristics (area and volume) as well as the constraint status of each forest component, are reported twice during each iteration. The first report occurs prior to harvesting, regeneration and growth processes, to indicate resource condition and the harvest candidate availability at the start of the period.

Note: Any transfers and floating group changes which occurred at the start of the period will be reflected in this report.

BEGIN **Report Inventory by zone, group and total landbase.**

END;

BEGIN **Report Constraints by zone and group (before):**

END;

3.6 Harvest

The harvest proceeds through the harvest candidate list. Each candidate class area is compared to the remaining harvest target. FSSIM calculates the maximum area that can be harvested from the class without exceeding any applicable targets, or violating any FCCs specified for any zones or groups to which the forest class is currently assigned. The portion of the class harvested is then assigned to the appropriate regenerated AU, and the data structure is updated to reflect state changes.

BEGIN **Harvest**
FOR EACH CandidateClass IN CurrentPeriod DO
BEGIN
Targets:
Calculate maximal area to harvest based on specified harvest targets.
Green-up:
Calculate maximal area to harvest based on green-up requirements.
Old-Growth:
Calculate maximal area to harvest based on old-growth requirements.
IF HarvestArea > 0 DO
BEGIN DoHarvest

When more than one target applies to a given class, the target that returns the least area is the one that applies. This ensures that no targets are exceeded.

Regenerate;
Regenerate harvested area. This entails assignment of regenerating AU values, and updating of the data structures. Multiple regenerating AU assignments can be specified in the regen.dat input file. In this case, the harvested area would be distributed across several AUs on this basis. Ages of regenerated classes are dependent upon the regen lag specified in the input file. A regen lag of +3 will cause the initial age to be set at -3 (0-3). Conversely, a regen lag of -3 sets the initial age at +3 (0-(-3)).
END DoHarvest;
ENDIF
END;
END ;

3.7 Report Constraints (After)

BEGIN **Report Constraints by zone and group (after):**
END ;

3.8 Road

Each class is assessed with respect to current roaded status. If the class age at the start of the analysis was above the maximum roaded age, and the class has not been previously roaded, its area is reduced by the roading percentage. Roaded Area is removed from zones and groups. This reduces the base area for the land unit, thereby affecting subsequent constraint calculations. This represents a compromise between considering roaded area as permanently non-green (too restrictive), and allowing it to green-up along with the regenerating components (too lax).

BEGIN **Road;**
    FOR EACH HarvestedClass DO
    BEGIN
        Update Group(Static/Float) to reflect roading.
        Update Zone to reflect roading.
        Update AU to reflect roading.
        Update CLASS to reflect roading.
    END;
END;

Note: Roading occurs after constraint reporting is completed as roading may change the zone or group land area, thereby affecting constraint calculations.

3.9 Aggregate

If the aggregation option is set to ON in the *config.dat* file, all like classes will be aggregated to minimize the size of the class list. Forest classes are considered similar if they match on:

age,
parent zone,

AU,
class type (and reversion age),
group membership,
roaded status, and
transfer criteria.

```
BEGIN Aggregate Post Harvest;
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR ALL ZONE associated AU  DO
                BEGIN
                        FOR ALL AU associated CLASS  DO
                        BEGIN
                                Aggregate like CLASSES
                        END;
                        Assign new aggregated CLASSLIST TO AU
                END;
        END;
END;
```

3.10 Age

Aging entails incrementing the age of each class by the period width.

```
BEGIN Age Post Harvest;
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR ALL ZONE associated AU  DO
                BEGIN
                        FOR ALL AU associated CLASS  DO
                        BEGIN
                                Age CLASS to period end
                        END;
                END;
        END;
END;
```

3.11 Populate (After)

After completing harvest and growth processes for the period, the FSSIM internal data structure is updated to reflect changes during the period. This is done for post-harvest reporting only, as the data structure will be updated again at the start of the next period to reflect transfers.

```
BEGIN Populate Groups
        FOR EACH Zone IN CurrentPeriod DO
        BEGIN
                FOR ZONE associated AU  DO
                BEGIN
                        FOR AU associated CLASS  DO
                        BEGIN
                                Update Static and Float Group structures for CLASS
                        END;
                END;
        END;
END;
```

3.12 Report (After)

```
BEGIN Report Area Roaded by zone and group
END ;

BEGIN Report Inventory (after):
END ;

BEGIN Report Harvest Levels
END ;

BEGIN Report Harvest Ages
END ;
```

APPENDIX I
INTERNAL DATA RECORD STRUCTURE

This appendix describes the critical FSSIM internal data record structures. The purpose in presenting this information is to provide the user with an understanding of the information contained within the internal data base, and the relationships between data components.

I.1 ZONE RECORD STRUCTURE 32
I.2 AGEAREA RECORD STRUCTURE 32
I.3 AXSDEF RECORD STRUCTURE (WITHIN AXSLIST) 32
I.4 TRANSFER RECORD STRUCTURE (TRANSFER) 32
I.5 TARGET RECORD STRUCTURE (TARGET) 33
I.6 ACHIEVED RECORD STRUCTURE (ACHIEVED) 33
I.7 ROADREC RECORD STRUCTURE (ROADREC) 33
I.8 INVENTORY REPORT RECORD STRUCTURE 33
I.9 AU RECORD STRUCTURE 34
I.10 REGEN RECORD STRUCTURE 34
I.11 GROUP RECORD STRUCTURE 34
I.12 GROUPDEF RECORD STRUCTURE (CLASS) 35
I.13 CLASS RECORD STRUCTURE 35
I.14 ROAD CLASS RECORD STRUCTURE 35

I.1 ZONE RECORD STRUCTURE

Type	Name	Description
int	id;	Zone ID number
float	area;	Area of Zone
int	*priority;	Harvest priority for Candidate List
AGEAREA	*agearea;	Age/area distributions (for FCC calculations)
LIST_PTR	*aulist;	Analysis Unit(see AU)
LIST_PTR	*axslist;	Access Constraints
LIST_PTR	*transferlist;	Transfer Information
TARGET	*target;	Target(area,volume)
ACHIEVED	*achieved;	Achieved levels(area,volume)
ROADREC	*roadrec;	Age of stand for roading and percent
INV_REPORT	*invrep;	Associated Inventory Report
int	complete;	Logical: True or False
float	total_area_roaded;	Total roaded area

I. 2 AGEAREA RECORD STRUCTURE

Type	Name	Description
int	indexlow;	Based on access constraints
int	indexhigh;	Based on access constraints
int	size;	
float	*data;	Variable

I.3 AXSDEF RECORD STRUCTURE (within AXSLIST)

Type	Name	Description
int	start_period;	Period start for access constraints
int	end_period;	Period end for access constraints
int	age1;	Minimum green-up age
float	pc1;	Maximum disturbance %
int	age2;	Minimum retention age
float	pc2;	Minimum retention %

I.4 TRANSFER RECORD STRUCTURE (Transfer)

Type	Name	Description
int	id;	Transfer ID
int	start_period;	Period to Transfer
int	end_period;	Period Transfer ends
ZONE	*zone;	Zone to Transfer
AU	*au;	AU to Transfer
GROUP	*group;	Group to Transfer

CLASS	*class;	Class to Transfer
int	age_low;	Low age
int	age_high;	High age
int	class_type;	Class Type
ZONE	*to_zone;	Zone to Transfer to
AU	*to_au;	AU to Transfer to
int	to_age;	Age to transfer to
int	to_class_type;	Class type to transfer to
int	to_rage;	Reversion age to transfer to

I.5 TARGET RECORD STRUCTURE (Target)

Type	Name	Description
float	*volume;	Target Volume to harvest
float	*area;	Target Area to harvest
float	*thin;	Not Used

I.6 ACHIEVED RECORD STRUCTURE (Achieved)

Type	Name	Description
float	volume;	Achieved volume
float	area;	Achieved area
float	thin;	Not used
float	road;	Roaded area
LIST_PTR	*activity;	Achieved activity by age (for harvest age reports)

I.7 ROADREC RECORD STRUCTURE (RoadRec)

Type	Name	Description
int	age;	Age Roaded
float	pc;	% Roaded

I.8 INVENTORY REPORT RECORD STRUCTURE

Type	Name	Description
float	*total_volume;	As Named
float	*total_area;	
float	*merch_volume;	
float	*merch_area;	
float	*available_area;	
float	*available_volume;	
float	*operable_area;	
float	*operable_volume;	
float	*inoperable_area;	
float	*inoperable_volume;	

```
float       *deferred_area;
float       *deferred_volume;
float       *noncontributing_area;
float       *noncontributing_volume;
```

I.9 AU RECORD STRUCTURE

Type	Name	Description
int	id;	AU ID
float	area;	Area of AU
float	landoaf;	Operational Adjustment Factor
int	*priority;	Harvest priority for candidate list
LIST_PTR	*classlist;	List of Classes that are with in this AU(see below CLASS)
LIST_PTR	*regenlist;	Associated Regen list(delay and %)
LIST_PTR	*transferlist;	Associated transfer list(see above TRANSFER)
TARGET	*target;	Target Info(see above TARGET)
ZONE	*parent_zone;	Zone that this AU is related to(see above ZONE)
ACHIEVED	*achieved;	Achieved Info(see above ACHIEVED)
int	complete;	Logical True or False
float	total_area_roaded;	Total roaded area

I.10 REGEN RECORD STRUCTURE

Type	Name	Description
AU	*au;	AU
int	delay;	Delay Years
float	pc;	Percent

I.11 GROUP RECORD STRUCTURE

Type	Name	Description
int	id;	Group ID
int	grouptype;	Group Type
float	area;	Area
int	*priority;	Harvest Priority
AGEAREA	*agearea;	Age/area distribution for FCC calculation
LIST_PTR	*axslist;	Access Constraint list
LIST_PTR	*transferlist;	Transfer list
LIST_PTR	*groupdefs;	Class
TARGET	*target;	Target(area,volume)
ACHIEVED	*achieved;	Achieved level
INV_REPORT	*invrep;	Associated Inventory report
int	complete;	Complete Flag(TRUE/FALSE)
float	total_area_roaded;	Total Area roaded

I.12 GROUPDEF RECORD STRUCTURE

Type	Name	Description
int	zone,	Zone ID
	au,	AU ID
	low_age,	Low age
	high_age,	High age
	class_type;	Class Type(Non-Contributing etc.)

I.13 CLASS RECORD STRUCTURE

Type	Name	Description
int	id;	Class ID
int	age;	Age of stand
int	class_type;	Class Type
int	reversion_age;	Reversion age
float	area;	Area of Class
ZONE	*parent_zone;	Related to Zone(see above)
AU	*parent_au;	Related to AU(see above)
TABLE	*table;	Associated operational adjustment factors
LIST_PTR	*floatgrouplist;	Float groups to which class belongs
LIST_PTR	*staticgrouplist;	Static groups to which class belongs
LIST_PTR	*transferlist;	Transfer Info(see above)
int	roaded;	Logical True or False(roaded?)
int	transferred;	Logical True or False(transferred?)
ROADCLASS	*roadclass;	Road Information

I.14 ROAD CLASS RECORD STRUCTURE

Type	Name	Description
ZONE	*zone;	Associated Zone(see above)
AU	*au;	Associated AU(see above)
LIST_PTR	*floatgrouplist;	Float group criteria(applies when condition met)
LIST_PTR	*staticgrouplist;	Static group criteria(applies always)
int	age;	age at roading
float	pc;	% roaded